



FACULTADE DE MATEMÁTICAS

Traballo Fin de Grao

# Comparación de modelos en la reducción de ruido de imágenes digitales bidimensionales

Pedro Meijide Suárez

2018/2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



GRAO DE MATEMÁTICAS

Traballo Fin de Grao

# Comparación de modelos en la reducción de ruido de imágenes digitales bidimensionales

Pedro Meijide Suárez

2019

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



# Trabajo propuesto

<b>Área de Coñecemento: Matemática aplicada</b>
<b>Título: Comparación de métodos en la reducción de ruido de imágenes digitales bidimensionales</b>
<b>Breve descripción do contido</b>
Se estudiarán desde el punto de vista teórico algunos algoritmos basados en minimización de funcionales para reducir el ruido de una imagen digital. Desde el punto de vista práctico, se elaborarán códigos para implementar dichos algoritmos para comparar su eficacia con diferentes modelos de ruido.
<b>Recomendacións</b>
Dominio de la teoría clásica de derivación numérica, el método de diferencias finitas para resolver ecuaciones diferenciales en derivadas parciales, y un lenguaje de programación (MatLab).



# Índice general

<b>Resumen</b>	<b>VIII</b>
<b>Introducción</b>	<b>XI</b>
<b>1. Ruido de una imagen y filtros para su eliminación</b>	<b>1</b>
1.1. Definición y tipos de ruido . . . . .	1
1.2. Algunos ejemplos de filtros para la eliminación de ruido . . . . .	3
1.3. Métodos variacionales para la eliminación de ruido . . . . .	5
<b>2. Un método variacional para eliminar ruido</b>	<b>7</b>
2.1. Planteamiento del problema . . . . .	7
2.2. Discretización del operador laplaciano . . . . .	8
2.3. Estudio del problema discreto . . . . .	12
<b>3. Resultados numéricos</b>	<b>21</b>
3.1. Ejemplo de comprobación . . . . .	21
3.2. Resultados con imágenes . . . . .	23
<b>Bibliografía</b>	<b>45</b>
<b>Apéndices</b>	<b>46</b>
<b>Programa ejemplo de comprobación</b>	<b>49</b>
<b>Programa para la eliminación de ruido</b>	<b>51</b>
<b>Ejemplo comprobación</b>	<b>55</b>
<b>Laplaciano discreto</b>	<b>57</b>
<b>Bilaplaciano discreto</b>	<b>59</b>

Funciones ejemplo
-------------------

63
----







## Resumen

En este trabajo estudiamos y comparamos algunos métodos variacionales para la eliminación de ruido en imágenes digitales. Empezamos definiendo el ruido y algunos tipos de éste, junto con algunos filtros básicos para su eliminación. A continuación presentamos un problema variacional basado en el operador laplaciano, que posteriormente estudiamos para su resolución numérica. Finalmente, resolvemos dicho problema y comparamos los resultados obtenidos con los de otro método para la reducción de ruido basado en el operador bilaplaciano.

## Abstract

In this work we study and compare some variational methods for the elimination of noise in digital images. We start by defining noise and some types of it, along with some basic filters for its elimination. Next, we present a variational problem based on the harmonic operator, which we later study for the sake of its numerical resolution. Finally, we solve this problem and compare the obtained results with those from another noise reduction method based on the biharmonic operator.



# Introducción

Una imagen digital es la representación bidimensional de una imagen como un conjunto rectangular de puntos llamados píxeles. Entonces, una imagen digital se puede representar como una tabla en la que la posición de cada elemento de ésta representa la localización del píxel y su valor es la intensidad de éste.

Desde el punto de vista matemático, una imagen también se puede pensar como una función de un dominio rectangular de  $\mathbb{R}^2$  en  $\mathbb{R}^c$  (si  $c = 1$ , estamos en el caso de una imagen en escala de grises, y si  $c = 3$ , estamos en el caso de imágenes RGB, las cuales definiremos a continuación). En este contexto, los métodos variacionales para reducir ruido en una imagen digital consisten en resolver un problema variacional cuyo segundo miembro aparece la imagen que contiene ruido.

Existen varios tipos de imágenes digitales, de los cuales los principales son:

- **Escala de grises:** en este tipo de imágenes la intensidad de los píxeles toma valores enteros entre 0 y  $2^n - 1$ , donde  $n$  es la profundidad en la escala de grises, el valor 0 corresponde al color negro, y el valor  $2^n - 1$  al blanco. Entonces, el número de intensidades posibles de la imagen es  $2^n$ . En el de  $n = 0$ , tenemos imágenes en blanco y negro, también llamadas imágenes binarias.
- **RGB:** corresponden a las siglas de rojo, verde y azul en inglés, y es la forma habitual de codificar las imágenes en color principalmente en pantallas. Estas imágenes se representan mediante tres tablas (una por color) llamadas canales, de modo que cada una de ellas sería análoga a una imagen en escala de grises.

También podríamos considerar imágenes médicas, imágenes de microscopio, imágenes de telescopio, y otras. No obstante, aquí solamente consideraremos imágenes de los dos tipos anteriormente descritos.



# Capítulo 1

## Ruido de una imagen y filtros para su eliminación

### 1.1. Definición y tipos de ruido

Llamamos ruido de una imagen a cualquier variación de su valor real que ésta puede tener, ya sea en la adquisición, transmisión o almacenamiento de la imagen. El ruido se caracteriza por no ser estéticamente agradable, de ahí el interés de querer eliminarlo, en caso de ser posible.

El ruido de una imagen se genera, principalmente, durante el proceso de digitalización, en el que influyen factores como el sensor de imagen, la intensidad de la luz, el enfoque, la codificación de la imagen... Por esto, utilizamos el procesamiento de imágenes para eliminar el ruido de una imagen digital y así poder obtener una imagen mas cercana a la original. Podemos simular una imagen digital con ruido de la manera siguiente:

$$D = O + R$$

donde  $D$  es la matriz que representa la imagen digital,  $O$  la que representa la imagen original y  $R$  la que representa el ruido. A partir de ahora consideraremos exclusivamente y sin pérdida de generalidad imágenes en escala de grises, pues los algoritmos aquí vistos podrían aplicarse a cada canal de una imagen RGB para la eliminación de ruido.

Definimos a continuación tres tipos distintos de ruido importantes en la práctica:

- **Ruido sal y pimienta:** consiste en la introducción aleatoria de blanco o negro puro en la imagen. Este tipo de ruido se puede generar en Matlab con el comando

```
I=imnoise(A,'salt & pepper',d)
```

donde  $d$  es el porcentaje de imagen a la que se le añade el ruido. Mostramos a continuación el efecto de este ruido en la Figura 1.1, tomada de [7].



Figura 1.1: Lenna en escala de grises y Lenna con ruido sal y pimienta.

- **Ruido gaussiano:** en este caso, la variación de ruido sigue una distribución normal. Este es el ruido más comúnmente utilizado en la simulación, porque lo describe aceptablemente en muchas situaciones prácticas, tales como el ruido procedente del sensor de la cámara. Se pueden generar imágenes con ruido gaussiano en MatLab mediante el comando

$$I = \text{imnoise}(A, 'gaussian', m, v)$$

donde  $m$  y  $v$  son la media y la varianza de la distribución normal del ruido considerado.



Figura 1.2: Lenna en escala de grises con ruido gaussiano.

- **Ruido uniforme:** a semejanza del ruido gaussiano, sigue otra distribución estadística conocida, que es la uniforme. Para generarlo en Matlab, se utiliza el comando

$$I = \text{imnoise}(A, 'speckle', v)$$

donde  $v$  es la varianza de la distribución.





Figura 1.3: Lenna en escala de grises con ruido uniforme.

Hay más distribuciones estadísticas que pueden utilizarse para simular ruido (ver, por ejemplo, [9]).

## 1.2. Algunos ejemplos de filtros para la eliminación de ruido

Veamos ahora algunos métodos elementales para la eliminación de ruido. Estos métodos consisten en cambiar los valores de cada píxel de la imagen  $D$  que representa la imagen con ruido, utilizando los píxeles vecinos con ese fin. A continuación, enunciaremos algunos ejemplos:

- **Media:** este filtro lineal consiste en cambiar el valor de un píxel por el promedio de sus píxeles vecinos. Este filtro suaviza la imagen, y por lo tanto conserva mal los bordes. No obstante, es útil a veces para preparar la imagen para otro tipo de manipulación.
- **Mediana:** este filtro consiste en sustituir el valor de un píxel por la mediana de los píxeles de su vecindad. Este filtro se beneficia de la robustez de la mediana como medida de tendencia central. Funciona muy bien con ruido en puntos aislados, y conserva muy bien los bordes, aunque frente a la presencia de mucho ruido, necesitamos mucha resolución para obtener buenos resultados.



Figura 1.4: Izquierda: ruido gaussiano. Centro: filtro media. Derecha: filtro mediana.



Figura 1.5: Izquierda: ruido sal y pimienta. Centro: filtro media. Derecha: filtro mediana.

- **Orden generalizado:** es una generalización del filtro de la mediana. Consiste en coger una vecindad de un píxel, ordenar los que la forman de menor a mayor, y escoger el valor de la posición  $n$ -ésima. Este filtro se aplica en Matlab mediante el comando

$$I = \text{ordfilt2}(A, \text{order}, \text{domain})$$

donde **order** es el valor que se toma de la vecindad **domain**. Para el caso particular de que **order** sea el mínimo, el máximo o la mediana, este filtro es el filtro máximo, el filtro mínimo o el filtro mediana.



Figura 1.6: Izquierda: ruido gaussiano. Centro: filtro máximo. Derecha: filtro mínimo.

- **Gaussiano:** este filtro es similar al de la media pero en este caso el peso de los píxeles de la vecindad vienen dados por la función gaussiana bidimensional dada por

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right),$$

donde  $\sigma$  es la desviación típica que queremos considerar.

### 1.3. Métodos variacionales para la eliminación de ruido

Estos métodos consisten en considerar la imagen con ruido  $f$  como una función  $f : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ , y asociarle un problema de optimización cuya solución es la imagen sin ruido. Por ejemplo, (véase [6]) consideramos el problema de hallar la imagen  $u$  que minimiza el funcional

$$E(u) = a\Phi(u) + b \int_{\Omega} (f - u)^2 dx,$$

donde  $a, b \in \mathbb{R}^+$  son constantes de ponderación. El segundo término es un término de fidelización, que nos asegura que la imagen  $u$  se parece a la imagen  $f$ . El primer término es un término regularizador, que intenta cancelar el ruido en la imagen  $u$ , penalizando su existencia, ya que el ruido produce mucha variación en la imagen. Algunos ejemplos para este término son

$$\Phi(u) = \int_{\Omega} (\Delta u)^2 dx, \tag{1.1}$$

$$\Phi(u) = \int_{\Omega} |\nabla u|^2 dx, \tag{1.2}$$

siendo  $\Delta, \nabla$  los operadores laplaciano y gradiente respectivamente.

El caso (1.1) ya fue estudiado en [8]. En el capítulo siguiente estudiaremos el caso (1.2), y veremos que conserva los bordes mejor que el método anterior. En el capítulo

tres, compararemos resultados obtenidos mediante ambos métodos, resaltando sus puntos fuertes y débiles.

## Capítulo 2

# Un método variacional para la eliminación de ruido

### 2.1. Planteamiento del problema

De acuerdo con [6], planteamos el siguiente problema variacional:

$$\left. \begin{aligned} &\text{Hallar } u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R} \text{ tal que minimiza el funcional} \\ &E(u) = \frac{\lambda}{2} \int_{\Omega} |\nabla u|^2 dx + \frac{1}{2} \int_{\Omega} (f - u)^2 dx \end{aligned} \right\}, \quad (2.1)$$

donde  $\lambda$  es un parámetro de ajuste, el término  $\int_{\Omega} (f - u)^2 dx$  es un término de fidelidad que nos asegura que  $u$  va a estar cerca de  $f$ , y  $\int_{\Omega} |\nabla u|^2 dx$  es un término regularizador.

Suponiendo que la solución de (2.1) es regular, entonces el mínimo debe ser solución de la ecuación en derivadas parciales de Euler-Lagrange, de acuerdo con [1]. Para llegar a dicha ecuación, basta tomar una función regular  $v(x, y)$  definida en  $\Omega$  y de soporte compacto, de modo que se anule en  $\partial\Omega$  y considerar la función de variable real  $i : \mathbb{R} \rightarrow \mathbb{R}$  definida como

$$i(\tau) = E(u + \tau v).$$

Por lo tanto, si  $u$  es solución de (2.1), tenemos que  $i$  tiene un mínimo relativo en  $\tau = 0$  y así

$$i'(0) = 0.$$

Desarrollando la expresión anterior tenemos:

$$i(\tau) = E(u + \tau v) = \frac{\lambda}{2} \int_{\Omega} \left( \left( \frac{\partial u}{\partial x} + \tau \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} + \tau \frac{\partial v}{\partial y} \right)^2 \right) dx + \frac{1}{2} \int_{\Omega} (f - u - \tau v)^2 dx.$$

Por lo tanto:

$$i'(\tau) = \frac{\lambda}{2} \int_{\Omega} \left( 2 \left( \frac{\partial u}{\partial x} + \tau \frac{\partial v}{\partial x} \right) \frac{\partial v}{\partial x} + 2 \left( \frac{\partial u}{\partial y} + \tau \frac{\partial v}{\partial y} \right) \frac{\partial v}{\partial y} \right) dx - \frac{1}{2} \int_{\Omega} 2(f - u - \tau v)v dx.$$

Evalutando la expresión anterior en el 0:

$$i'(0) = \lambda \int_{\Omega} \left( \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) dx - \int_{\Omega} (f - u)v dx.$$

Ahora, aplicando la fórmula de Green (ver, por ejemplo, [4]), y dado que  $v$  se anula en  $\partial\Omega$ , tenemos que:

$$\int_{\Omega} \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} dx = - \int_{\Omega} \frac{\partial^2 u}{\partial x^2} v dx + \int_{\partial\Omega} \frac{\partial u}{\partial x} v n_1 d\Gamma = - \int_{\Omega} \frac{\partial^2 u}{\partial x^2} v dx,$$

$$\int_{\Omega} \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} dx = - \int_{\Omega} \frac{\partial^2 u}{\partial y^2} v dx + \int_{\partial\Omega} \frac{\partial u}{\partial y} v n_2 d\Gamma = - \int_{\Omega} \frac{\partial^2 u}{\partial y^2} v dx,$$

donde  $n = (n_1, n_2)$  es el vector normal orientado positivamente a lo largo de  $\partial\Omega$  y  $d\Gamma$  es el elemento diferencial de la medida en  $\partial\Omega$ .

Así, tenemos que, para toda función  $v(x, y)$  en las condiciones anteriores se verifica:

$$0 = i'(0) = \lambda \int_{\Omega} \left( \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} \right) dx = -\lambda \int_{\Omega} \Delta u v dx - \int_{\Omega} (f - u)v dx,$$

cuya formulación fuerte es:

$$-\lambda \Delta u + u - f = 0.$$

Finalmente, podemos reescribir el problema variacional (2.1) como el siguiente problema en ecuaciones en derivadas parciales:

$$\left. \begin{array}{l} \text{Hallar } u : \Omega \rightarrow \mathbb{R} \text{ tal que} \\ -\lambda \Delta u + u = f \text{ en } \Omega \\ u = 0 \text{ sobre } \partial\Omega \end{array} \right\}. \quad (2.2)$$

## 2.2. Obtención de una discretización del operador laplaciano invariante por rotaciones

Con el fin de minimizar el impacto de la reducción de ruido en la imagen cuando en ésta hay objetos similares girados en distintos ángulos, vamos a buscar entonces una discretización del laplaciano que sea invariante por rotaciones. Para ello, nos inspiraremos en [2] en lo que sigue.

A partir de ahora, tomaremos el dominio  $\Omega$  como un rectángulo de la forma  $[a, b] \times [c, d]$ , lo que nos permitirá considerar una discretización en  $x$  e  $y$  de la forma siguiente:

$$a = x_0 < x_1 < \dots < x_i < \dots < x_{N-1} < x_N = b, \quad x_i = x_0 + ih, \quad i = 0, 1, \dots, N$$

$$c = y_0 < y_1 < \dots < y_j < \dots < y_{M-1} < y_M = d, \quad y_j = y_0 + jh, \quad j = 0, 1, \dots, M$$

y denotaremos  $u(x_i, y_j) = U_{ij}$ .

Recordemos la fórmula unidimensional centrada de orden 2 centrada para la derivada segunda siguiente.

$$f''(\alpha) = \frac{f(\alpha + h) - 2f(\alpha) + f(\alpha - h)}{h^2} + O(h^2). \quad (2.3)$$

Aplicando (2.3) al operador laplaciano

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2},$$

llegamos a la primera aproximación de las derivadas parciales de  $u$ :

$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{U_{i-1,j} - 2U_{ij} + U_{i+1,j}}{h^2} + O(h^2),$$

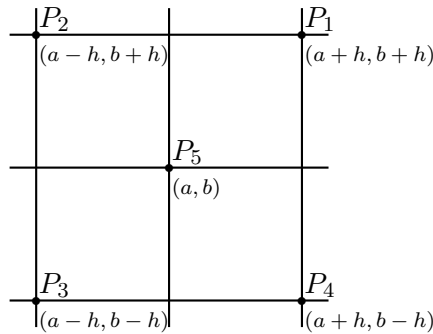
$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{U_{i,j-1} - 2U_{ij} + U_{i,j+1}}{h^2} + O(h^2).$$

Entonces, juntando las dos aproximaciones anteriores, llegamos a la aproximación numérica de  $\Delta u$  dada por

$$\Delta u(x_i, y_j) \approx \frac{U_{i,j-1} + U_{i-1,j} - 4U_{ij} + U_{i,j+1} + U_{i+1,j}}{h^2}, \quad (2.4)$$

con orden 2 en  $x$  e  $y$ . Nos referiremos a la discretización anterior como el laplaciano discreto de cinco puntos. Es inmediato comprobar que (2.4) no es invariante por rotaciones.

En consecuencia, vamos a buscar ahora otra aproximación del laplaciano discreto, pero que ésta sea invariante por rotaciones. Con el fin de obtenerla, buscaremos una segunda aproximación del laplaciano para combinarla con la que acabamos de obtener. Un paso previo es el siguiente teorema (en el que denotaremos las derivadas parciales con subíndices por comodidad):



**Teorema 2.1.** *Sea  $S$  un rectángulo abierto que contenga a los  $P_1, \dots, P_5$  puntos del esquema y sea  $\omega : S \rightarrow \mathbb{R}$  una función de clase 4 en  $S$ . Entonces existen  $c_1, \dots, c_5$  escalares tales que*

$$\sum_{i=1}^5 c_i \omega(P_i) = h^2(\omega_{xx}(P_5) + \omega_{yy}(P_5)) + O(h^2)$$

*Demostración.* Sabemos que el desarrollo de Taylor de de orden 4 de una función suficientemente regular de dos variables centrado en el punto  $(a + h, b + k)$  viene dado por:

$$\begin{aligned} \omega(a + h, b + k) &= \omega(a, b) + h\omega_x(a, b) + k\omega_y(a, b) \\ &\quad + \frac{1}{2!}(h^2\omega_{xx}(a, b) + 2kh\omega_{xy}(a, b) + k^2\omega_{yy}(a, b)) \\ &\quad + \frac{1}{3!}(h^3\omega_{xxx}(a, b) + h^2k\omega_{xxy}(a, b) + hk^2\omega_{xyy}(a, b) + k^3\omega_{yyy}(a, b)) + O(h^l k^{4-l}) \end{aligned}$$

con  $l = 1, \dots, 4$  un índice de suma.

Por lo tanto, tenemos que:

$$\begin{aligned} \omega(P_1) &= \omega(a, b) + h(\omega_x(a, b) + \omega_y(a, b)) + \frac{h^2}{2!}(\omega_{xx}(a, b) + 2\omega_{xy}(a, b) + \omega_{yy}(a, b)) \\ &\quad + \frac{h^3}{3!}(\omega_{xxx}(a, b) + \omega_{xxy}(a, b) + \omega_{xyy}(a, b) + \omega_{yyy}(a, b)) + O(h^4), \\ \omega(P_2) &= \omega(a, b) + h(-\omega_x(a, b) + \omega_y(a, b)) + \frac{h^2}{2!}(\omega_{xx}(a, b) - 2\omega_{xy}(a, b) + \omega_{yy}(a, b)) \\ &\quad + \frac{h^3}{3!}(-\omega_{xxx}(a, b) + \omega_{xxy}(a, b) - \omega_{xyy}(a, b) + \omega_{yyy}(a, b)) + O(h^4), \\ \omega(P_3) &= \omega(a, b) - h(\omega_x(a, b) + \omega_y(a, b)) + \frac{h^2}{2!}(\omega_{xx}(a, b) + 2\omega_{xy}(a, b) + \omega_{yy}(a, b)) \\ &\quad - \frac{h^3}{3!}(\omega_{xxx}(a, b) + \omega_{xxy}(a, b) + \omega_{xyy}(a, b) + \omega_{yyy}(a, b)) + O(h^4), \\ \omega(P_4) &= \omega(a, b) + h(\omega_x(a, b) - \omega_y(a, b)) + \frac{h^2}{2!}(\omega_{xx}(a, b) - 2\omega_{xy}(a, b) + \omega_{yy}(a, b)) \\ &\quad + \frac{h^3}{3!}(\omega_{xxx}(a, b) - \omega_{xxy}(a, b) + \omega_{xyy}(a, b) - \omega_{yyy}(a, b)) + O(h^4), \\ \omega(P_5) &= \omega(a, b). \end{aligned}$$

Sustituyendo en la igualdad e identificando los coeficientes de la misma potencia, llegamos al sistema de ecuaciones lineales siguiente:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ h & -h & -h & h & 0 \\ h & h & -h & -h & 0 \\ \frac{h^2}{2} & \frac{h^2}{2} & \frac{h^2}{2} & \frac{h^2}{2} & 0 \\ h^2 & -h^2 & h^2 & -h^2 & 0 \\ \frac{h^2}{2} & \frac{h^2}{2} & \frac{h^2}{2} & \frac{h^2}{2} & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ h^2 \\ 0 \\ h^2 \end{pmatrix},$$



donde en cada columna están los términos asociados a las funciones  $\omega, \omega_x, \omega_x, \omega_{xx}, \omega_{xy}, \omega_{yy}$ , respectivamente, de las aproximaciones de Taylor de  $\omega(P_i)$ , siendo  $i$  el número de la columna. Eliminando la última ecuación por ser redundante con la antepenúltima, y como  $h \neq 0$ , el sistema nos queda así:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 0 \\ 1 & 1 & -1 & -1 & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ 1 & -1 & 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}.$$

Resolviendo el sistema, obtenemos los valores

$$c_1 = \frac{1}{2}, c_2 = \frac{1}{2}, c_3 = \frac{1}{2}, c_4 = \frac{1}{2}, c_5 = -2.$$

Ahora ya solo nos queda comprobar que los términos correspondientes a las derivadas de orden 3 se anulan en el desarrollo de Taylor. En efecto:

$$\begin{aligned} \omega_{xxx} &: \frac{h^3}{3!}(c_1 - c_2 - c_3 + c_4) = 0, \\ \omega_{xxy} &: \frac{h^3}{3!}(c_1 + c_2 - c_3 - c_4) = 0, \\ \omega_{xyy} &: \frac{h^3}{3!}(c_1 - c_2 - c_3 + c_4) = 0, \\ \omega_{yyy} &: \frac{h^3}{3!}(c_1 + c_2 - c_3 - c_4) = 0. \end{aligned}$$

Y así queda demostrado el resultado.  $\square$

Una vez demostrado el anterior teorema, podemos escribir la nueva aproximación de  $\Delta u(x_i, y_j)$  como

$$\Delta u(x_i, y_j) \approx \frac{-4U_{ij} + U_{i+1,j+1} + U_{i+1,j-1} + U_{i-1,j-1} + U_{i-1,j+1}}{2h^2}. \quad (2.5)$$

A partir de ahora, denotaremos por  $(\Delta_h u_h)_{ij}$  la aproximación (2.4) y por  $(\bar{\Delta}_h u_h)_{ij}$  la aproximación (2.5). Ahora, definimos los vectores infinitos  $v_h, w_h$  como sigue:

$$v_{ij} = \begin{cases} 1, & \text{si } i > 0 \\ 0, & \text{en otro caso} \end{cases} \quad w_{ij} = \begin{cases} 1, & \text{si } i > j \\ 0, & \text{en otro caso} \end{cases}$$

Claramente, se ve que los vectores son el mismo pero rotado un ángulo de  $\frac{\pi}{4}$ , y es fácil comprobar que  $(\Delta_h v_h)_{00} = h^{-2}$ , y que  $(\Delta_h w_h)_{00} = 2h^{-2}$ , por lo tanto, hemos comprobado que el primer laplaciano discreto no es una discretización invariante por rotaciones.

Consideremos ahora una nueva discretización del laplaciano como combinación convexa de las dos anteriores, con  $\alpha$  un parámetro entre cero y uno, dada por

$$\Delta_h^{(r)} = \alpha \Delta_h + (1 - \alpha) \bar{\Delta}_h.$$

Para que la discretización anterior sea invariante por rotaciones, debe verificarse que  $(\Delta_h^{(r)} v_h)_{00} = (\Delta_h^{(r)} w_h)_{00}$ . Es fácil comprobar que entonces debe ser  $\alpha = \frac{1}{3}$ . Así obtenemos la nueva discretización siguiente:

$$\Delta u(x_i, y_j) \approx \frac{-4U_{ij} + U_{i+1,j} + U_{i,j+1} + U_{i,j-1} + U_{i-1,j}}{3h^2} + \frac{-4U_{ij} + U_{i+1,j+1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i-1,j-1}}{3h^2}. \quad (2.6)$$

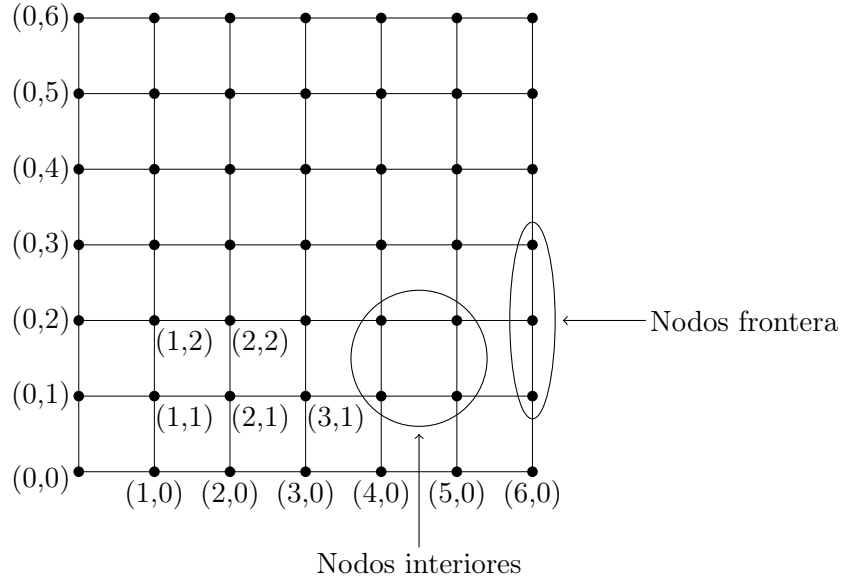


Figura 2.1: Ejemplo de problema discreto.

Nos referiremos ahora a la discretización anterior como laplaciano discreto invariante por rotaciones.

### 2.3. Estudio del problema discreto

Definimos ahora el vector  $\tilde{u}$  como el vector:

$$\tilde{u} = [U_{11} \ U_{21} \ \dots \ U_{N1} \ U_{12} \ \dots \ U_{N2} \ \dots \ U_{NM}],$$

y el operador lineal  $L_h: \mathbb{R}^{n \times M} \rightarrow \mathbb{R}^{N \times M}$  dado por:

$$(L_h \tilde{u})_{(i,j)} = \frac{-4U_{ij} + U_{i+1,j} + U_{i,j+1} + U_{i,j-1} + U_{i-1,j}}{3h^2} + \frac{-4U_{ij} + U_{i+1,j+1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i-1,j-1}}{3h^2},$$

cuya matriz denotaremos también como  $L_h$  sin ningún tipo de ambigüedad.

Para el nodo  $(1, 1)$ :

$$(L_h \tilde{u})_{(1,1)} 3h^2 = -8U_{1,1} + U_{2,2} + U_{2,1} + U_{2,0} + U_{1,2} + U_{1,0} + U_{0,2} + U_{0,1} + U_{0,0}.$$

Por tener una condición de contorno tipo Dirichlet homogénea, tenemos que

$$(L_h \tilde{u})_{(1,1)} = -8U_{1,1} + U_{2,2} + U_{2,1} + U_{1,2}$$

que nos da los elementos de la primera fila de  $L_h$  siguientes:

$$\left( \overbrace{-8 \ 1 \ 0 \ \dots \ 0}^N \mid 1 \ 1 \ 0 \ \dots \ 0 \mid 0 \ \dots \ 0 \right).$$

Para un nodo de la forma  $(i, 1)$ , con  $1 \leq i \leq N-1$ :

$$(L_h \tilde{u})_{(i,1)} 3h^2 = -8U_{i,1} + U_{i+1,2} + U_{i+1,1} + U_{i,2} + U_{i-1,2} + U_{i-1,1},$$

que nos da los elementos de la fila  $i$ -ésima

$$\left( \overbrace{0 \ \dots \ 0}^{i-2} \ 1 \ -8 \ 1 \ 0 \ \dots \ 0 \mid \overbrace{0 \ \dots \ 0}^{i-2} \ 1 \ 1 \ 1 \ 0 \ \dots \ 0 \mid 0 \ \dots \ 0 \right).$$

Y para un nodo de la forma  $(N, 1)$ :

$$(L_h \tilde{u})_{(N,1)} = -8U_{N,1} + U_{N,2} + U_{N-1,2} + U_{N-1,1},$$

que nos da los elementos de la fila  $N$ -ésima siguientes:

$$\left( \overbrace{0 \ \dots \ 0 \ 1 \ -8}^N \mid 0 \ \dots \ 0 \ 1 \ 1 \mid 0 \ \dots \ 0 \right).$$

Vamos ahora a ver la forma de la última fila de bloques de la matriz  $L_h$ . Empecemos con el nodo  $(1, M)$ :

$$(L_h \tilde{u})_{(1,M)} = -8U_{1,M} + U_{1,M-1} + U_{2,M} + U_{2,M-1},$$

y así tenemos la fila siguiente con los elementos de la matriz:

$$(0 \ \dots \ 0 \mid 1 \ 1 \ 0 \ \dots \ 0 \mid \overbrace{-8 \ 1 \ 0 \ \dots \ 0}^N).$$

Para un nodo de la forma  $(i, M)$ , con  $1 \leq i \leq N - 1$ :

$$(L_h \tilde{u})_{(i,M)} 3h^2 = -8U_{i,M} + U_{i+1,M-1} + U_{i+1,M} + U_{i,M-1} + U_{i-1,M-1} + U_{i-1,M},$$

que nos da los elementos siguientes:

$$(0 \dots 0 | \overbrace{0 \dots 0}^{i-2} 1 1 1 0 \dots 0 | \overbrace{0 \dots 0}^{i-2} 1 - 8 1 0 \dots 0).$$

Y para el nodo  $(N, M)$ :

$$(L_h \tilde{u})_{(N,M)} 3h^2 = -8U_{N,M} + U_{N,M-1} + U_{N-1,M-1} + U_{N-1,M},$$

que nos da los elementos:

$$(0 \dots 0 | 0 \dots 0 1 1 | \overbrace{0 \dots 0 1}^N - 8).$$

Por último, veamos la forma de las filas intermedias. Tomando  $j$  de modo que  $1 \leq j \leq M - 1$ , veamos cómo son los nodos de la forma  $(i, j)$  con  $0 \leq i \leq N$ . Para el nodo  $(1, j)$ :

$$(L_h \tilde{u})_{(1,j)} 3h^2 = -8U_{1,j} + U_{2,j+1} + U_{2,j} + U_{2,j-1} + U_{1,j+1} + U_{1,j-1},$$

que nos da los elementos:

$$(\overbrace{0 0 \dots 0 0}^{(i-2) \text{ matrices}} | 1 1 0 \dots 0 | - 8 1 0 \dots 0 | 1 1 0 \dots 0 | 0 \dots 0).$$

Para un nodo de la forma  $(i, j)$  con  $1 \leq i \leq N - 1$ :

$$(L_h \tilde{u})_{(i,j)} 3h^2 = -8U_{ij} + U_{i+1,j+1} + U_{i+1,j} + U_{i+1,j-1} + U_{i,j+1} + U_{i,j-1} + U_{i-1,j+1} + U_{i-1,j} + U_{i-1,j-1},$$

y nos da los elementos:

$$(\overbrace{0 0 \dots 0 0}^{(i-2) \text{ matrices}} | \overbrace{0 \dots 0}^{i-2} 1 1 1 0 \dots 0 | \overbrace{0 \dots 0}^{i-2} 1 - 8 1 0 \dots 0 | \overbrace{0 \dots 0}^{i-2} 1 1 1 0 \dots 0 | 0 \dots 0).$$

Y por último, para el nodo  $(N, j)$ :

$$(L_h \tilde{u})_{(N,j)} 3h^2 = -8U_{N,j} + U_{N-1,j+1} + U_{N-1,j} + U_{N-1,j-1} + U_{N,j+1} + U_{N,j-1},$$

que nos da los elementos:

$$(\overbrace{0 0 \dots 0 0}^{(i-2) \text{ matrices}} | 0 \dots 0 1 1 | 0 \dots 0 1 - 8 | 0 \dots 0 1 1 | 0 \dots 0)$$

Por lo tanto, la matriz asociada a a la discretización dada por (2.6) es de la forma:

$$L_h = \frac{1}{3h^2} \begin{bmatrix} T & C & 0 & 0 & \cdots & 0 \\ C & T & C & 0 & \cdots & 0 \\ 0 & C & T & C & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & C & T & C \\ 0 & \cdots & & 0 & 0 & C & T \end{bmatrix}_{NM \times NM}$$

donde  $T$  y  $C$  son las matrices siguientes:

$$T = \begin{bmatrix} -8 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -8 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -8 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & -8 & 1 \\ 0 & \cdots & & 0 & 0 & 1 & -8 \end{bmatrix}_{N \times N}$$

$$C = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & 1 & 1 & 1 \\ 0 & \cdots & & 0 & 0 & 1 & 1 \end{bmatrix}_{N \times N}.$$

Veamos ahora cómo son los autovalores de la matriz asociada a la discretización (2.6) invariante por rotaciones, dada por la matriz que denotaremos por  $\Delta_h$  y que adopta la forma siguiente:

$$\Delta_h = \frac{1}{h^2} \begin{bmatrix} T & I & 0 & 0 & \cdots & 0 \\ I & T & I & 0 & \cdots & 0 \\ 0 & I & T & I & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & & 0 & I & T & I \\ 0 & \cdots & & 0 & 0 & I & T \end{bmatrix}_{NM \times NM}$$

donde  $T$  es la matriz descrita anteriormente, e  $I$  es la matriz identidad. Para calcular los autovalores de la matriz, adaptando la técnica de [3] para el laplaciano discreto, probaremos el siguiente resultado:

**Proposición 2.2.** *Dada una matriz tridiagonal*

$$B = \begin{pmatrix} a & b & & & \\ c & a & b & & \\ & \ddots & \ddots & \ddots & \\ & & c & a & b \\ & & & c & b \end{pmatrix}_{N \times N}$$

con  $a, b \neq 0$ ,  $c \neq 0$  con  $\frac{c}{b} > 0$ , sus  $N$  autovalores vienen dados por

$$\lambda_k = a + b \left( \frac{c}{b} \right)^{\frac{1}{2}} 2 \cos \left( \frac{\pi k}{N+1} \right),$$

con  $k = 1, \dots, N$

*Demostración.* Sea  $v$  un autovector con componentes  $v = (v_1, \dots, v_N)$ . Recordemos que los autovalores de la matriz  $B$  son los números reales  $\lambda$  que verifican  $Bv = \lambda v$ . Dicha ecuación se puede reescribir como:

$$\begin{aligned} (a - \lambda)v_1 + bv_2 &= 0, \\ cv_1 + (a - \lambda)v_2 + bv_3 &= 0, \\ &\dots \\ cv_{j-1} + (a - \lambda)v_j + bv_{j+1} &= 0, \\ &\dots \\ cv_{N-1} + (a - \lambda)v_N &= 0, \end{aligned}$$

que podemos escribir como la siguiente ecuación en diferencias:

$$cv_{j-1} + (a - \lambda)v_j + bv_{j+1} = 0, \quad j = 1, \dots, N,$$

tomando  $v_0 = v_{N+1} = 0$ . Podemos escribir la anterior ecuación en la forma matricial siguiente

$$V_j = AV_{j-1}, \quad j = 1, \dots, N,$$

donde  $V_j = \begin{pmatrix} v_{j+1} \\ v_j \end{pmatrix}$  y  $A = \begin{pmatrix} \frac{\lambda-a}{b} & -\frac{c}{b} \\ 1 & 0 \end{pmatrix}$ . En consecuencia, podemos escribir

$$V_j = A^j V_0, \quad j = 1, \dots, N. \quad (2.7)$$

Estudiemos cómo son las potencias de la matriz  $A$  en función de sus autovalores. Los autovalores de la matriz  $A$  vienen dados por las soluciones de la ecuación

$$\left(\frac{\lambda - a}{b} - m\right)(-m) + \frac{c}{b} = 0,$$

es decir,

$$m^2 + \frac{a - \lambda}{b}m + \frac{c}{b} = 0.$$

- Caso 1.- La ecuación anterior tiene dos raíces iguales. Esto es equivalente a que  $\left(\frac{a - \lambda}{b}\right) - 4\frac{c}{b} = 0$  y así la matriz  $A$  adopta la forma

$$A = \begin{pmatrix} \frac{\lambda - a}{b} & -\frac{(\lambda - a)^2}{4b^2} \\ 1 & 0 \end{pmatrix}.$$

Veamos ahora la forma de las potencias de  $A$ :

$$A^2 = \begin{pmatrix} \frac{3}{4}\left(\frac{\lambda - a}{b}\right)^2 & -\frac{1}{4}\left(\frac{\lambda - a}{b}\right)^3 \\ \frac{\lambda - a}{b} & -\frac{1}{4}\left(\frac{\lambda - a}{b}\right)^2 \end{pmatrix}, A^3 = \begin{pmatrix} \frac{1}{2}\left(\frac{\lambda - a}{b}\right)^3 & -\frac{3}{16}\left(\frac{\lambda - a}{b}\right)^4 \\ \frac{3}{4}\left(\frac{\lambda - a}{b}\right)^2 & -\frac{1}{4}\left(\frac{\lambda - a}{b}\right)^3 \end{pmatrix}.$$

Visto lo que ocurre para las potencias 2 y 3 de  $A$ , intentemos probar por inducción que

$$A^j = \begin{pmatrix} \frac{j+1}{2^j}\left(\frac{\lambda - a}{b}\right)^j & -\frac{j}{2^{j+1}}\left(\frac{\lambda - a}{b}\right)^{j+1} \\ \frac{j}{2^{j-1}}\left(\frac{\lambda - a}{b}\right)^{j-1} & -\frac{j-1}{2^j}\left(\frac{\lambda - a}{b}\right)^j \end{pmatrix}. \quad (2.8)$$

En efecto,

$$\begin{aligned} A^j A &= \begin{pmatrix} \frac{j+1}{2^j}\left(\frac{\lambda - a}{b}\right)^j \frac{\lambda - a}{b} - \frac{j}{2^{j+1}}\left(\frac{\lambda - a}{b}\right)^{j+1} & \frac{j+1}{2^j}\left(\frac{\lambda - a}{b}\right)^j \left(-\frac{(\lambda - a)^2}{4b^2}\right) \\ \frac{j}{2^{j-1}}\left(\frac{\lambda - a}{b}\right)^{j-1} \frac{\lambda - a}{b} - \frac{j-1}{2^j}\left(\frac{\lambda - a}{b}\right)^j & \frac{j}{2^{j-1}}\left(\frac{\lambda - a}{b}\right)^{j-1} \left(-\frac{(\lambda - a)^2}{4b^2}\right) \end{pmatrix} = \\ &= \begin{pmatrix} \frac{j+2}{2^{j+1}}\left(\frac{\lambda - a}{b}\right)^{j+1} & -\frac{j+1}{2^{j+2}}\left(\frac{\lambda - a}{b}\right)^{j+2} \\ \frac{j+1}{2^j}\left(\frac{\lambda - a}{b}\right)^j & -\frac{j}{2^{j+1}}\left(\frac{\lambda - a}{b}\right)^{j+1} \end{pmatrix} \end{aligned}$$

y (2.8) queda demostrado. Finalmente, usando (2.8), tenemos que:

$$\begin{pmatrix} v_{j+1} \\ v_j \end{pmatrix} = \begin{pmatrix} \frac{j+1}{2^j}\left(\frac{\lambda - a}{b}\right)^j & -\frac{j}{2^{j+1}}\left(\frac{\lambda - a}{b}\right)^{j+1} \\ \frac{j}{2^{j-1}}\left(\frac{\lambda - a}{b}\right)^{j-1} & -\frac{j-1}{2^j}\left(\frac{\lambda - a}{b}\right)^j \end{pmatrix} \begin{pmatrix} v_1 \\ v_0 \end{pmatrix}, \quad j = 1, \dots, N$$

y en consecuencia

$$v_j = \frac{j}{2^{j-1}} \left( \frac{\lambda - a}{b} \right)^j v_1, j = 1, \dots, N.$$

Por ser  $v_0 = v_{N+1} = 0$ , tenemos que  $v = 0$ , lo cual es un absurdo puesto que  $v$  es un autovector y éste no puede ser nulo.

- Caso 2.– La ecuación en diferencias tiene dos raíces distintas  $m_1, m_2$ , y por lo tanto la matriz  $A$  se puede diagonalizar de la forma siguiente:

$$P^{-1}AP = \begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix},$$

y tenemos que

$$A^j = P \begin{pmatrix} m_1^j & 0 \\ 0 & m_2^j \end{pmatrix} P^{-1}.$$

Usando (2.7), podemos escribir

$$v_j = Bm_1^j + Cm_2^j, j = 1, \dots, N$$

con  $B$  y  $C$  constantes. Por ser  $v_0 = v_{N+1} = 0$ , tenemos

$$\begin{aligned} B + C &= 0, \\ Bm_1^{N+1} + Cm_2^{N+1} &= 0, \end{aligned}$$

de donde se deduce que  $\left( \frac{m_1}{m_2} \right)^{N+1} = 1$  y como  $m_1 \neq m_2$ , tenemos que

$$\frac{m_1}{m_2} = e^{i \frac{2k\pi}{N+1}}.$$

Como  $m_1 m_2 = \frac{c}{b}$ , tenemos que

$$\begin{cases} m_1 = \left( \frac{c}{b} \right)^{1/2} e^{\frac{ik\pi}{N+1}} \\ m_2 = \left( \frac{c}{b} \right)^{1/2} e^{\frac{-ik\pi}{N+1}} \end{cases}, \text{ o bien } \begin{cases} m_1 = - \left( \frac{c}{b} \right)^{1/2} e^{\frac{ik\pi}{N+1}} \\ m_2 = - \left( \frac{c}{b} \right)^{1/2} e^{\frac{-ik\pi}{N+1}} \end{cases}, \text{ con } k = 1, \dots, N.$$

Por ser otra raíz de la unidad distinta de 1, las dos expresiones equivalen. Considerando el primer caso, tenemos que

$$m_1 + m_2 = \left( \frac{c}{b} \right)^{\frac{1}{2}} 2 \cos \left( \frac{\pi k}{N+1} \right),$$



y como  $(m - m_1)(m - m_2) = m^2 - \frac{\lambda - a}{b}m + \frac{c}{b}$  llegamos a que

$$\lambda = a + b \left( \frac{c}{b} \right)^{\frac{1}{2}} 2 \cos \left( \frac{\pi k}{N+1} \right),$$

y por lo tanto tenemos que los  $N$  autovalores de la matriz  $B$  son efectivamente

$$\lambda_k = a + b \left( \frac{c}{b} \right)^{\frac{1}{2}} 2 \cos \left( \frac{\pi k}{N+1} \right).$$

□

Para el caso de la matriz  $\Delta_h$ , tenemos que los autovalores de  $T$  son

$$\lambda_k = -4 + 2 \cos \left( \frac{\pi k}{N+1} \right), \quad k = 1, \dots, N.$$

Para calcular los autovalores de la matriz, utilizamos un resultado que podemos encontrar en [3], que dice lo siguiente:

**Teorema 2.3.** *Sea  $A$  una matriz por bloques que se puede escribir como*

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,m} \\ A_{2,1} & A_{2,2} & \cdots & A_{2,m} \\ \vdots & & & \vdots \\ A_{m,1} & A_{m,2} & \cdots & A_{m,m} \end{pmatrix},$$

donde cada  $A_{i,j}$  es una matriz cuadrada  $n \times n$ , y todas las matrices tienen un conjunto común de  $n$  autovectores linealmente independientes, entonces los autovalores de  $A$  vienen dados por los autovalores de las matrices

$$\begin{pmatrix} \lambda_{1,1}^{(k)} & \lambda_{1,2}^{(k)} & \cdots & \lambda_{1,m}^{(k)} \\ \lambda_{2,1}^{(k)} & \lambda_{2,2}^{(k)} & \cdots & \lambda_{2,m}^{(k)} \\ \vdots & & & \vdots \\ \lambda_{m,1}^{(k)} & \lambda_{m,2}^{(k)} & \cdots & \lambda_{m,m}^{(k)} \end{pmatrix}, \quad k = 1, \dots, n,$$

donde cada  $\lambda_{i,j}^{(k)}$  es el  $k$ -ésimo autovalor asociado al  $k$ -ésimo autovector común de las matrices  $A_{i,j}$ .

Puesto que se puede tomar cualquier vector como autovector asociado a un autovalor de la matriz identidad, estamos bajo las condiciones del teorema anterior. Por lo tanto,

vamos a calcular los valores propios de la matriz formada por los autovalores, que adopta la forma siguiente

$$\left[ \begin{array}{ccccccc} -\frac{4}{h^2} + \frac{2}{h^2} \cos\left(\frac{\pi k}{M+1}\right) & \frac{1}{h^2} & & & & & \\ & \frac{1}{h^2} & & & & & \\ & & \ddots & & & & \\ & & & \ddots & & & \\ & & & & \frac{1}{h^2} & & \\ & & & & & -\frac{4}{h^2} + \frac{2}{h^2} \cos\left(\frac{\pi k}{M+1}\right) & \end{array} \right]_{M \times M} \quad k = 1, \dots, N.$$

Como es una matriz tridiagonal, los autovalores quedan de la forma

$$\lambda_{k,s} = -\frac{4}{h^2} + \frac{2}{h^2} \cos\left(\frac{\pi k}{M+1}\right) + \frac{2}{h^2} \cos\left(\frac{\pi s}{M+1}\right), \quad k = 1, \dots, N, \quad s = 1, \dots, M.$$

Teniendo en cuenta que  $\cos(2\theta) = 1 - \sin^2 \theta$ ,

$$\begin{aligned} \cos\left(\frac{\pi k}{N+1}\right) &= 1 - \sin^2\left(\frac{\pi k}{2(N+1)}\right), \quad k = 1, \dots, N, \\ \cos\left(\frac{\pi s}{M+1}\right) &= 1 - \sin^2\left(\frac{\pi s}{2(M+1)}\right), \quad s = 1, \dots, M. \end{aligned}$$

Y por lo tanto, tenemos que

$$\begin{aligned} \lambda_{k,s} &= -\frac{4}{h^2} + \frac{2}{h^2} - \frac{2}{h^2} \sin^2\left(\frac{\pi k}{2(N+1)}\right) + \frac{2}{h^2} - \frac{2}{h^2} \sin^2\left(\frac{\pi s}{2(M+1)}\right) = \\ &= -\frac{4}{h^2} \left( \sin^2\left(\frac{\pi k}{2(N+1)}\right) + \sin^2\left(\frac{\pi s}{2(M+1)}\right) \right), \quad k = 1, \dots, N, \quad s = 1, \dots, M. \end{aligned}$$

Entonces  $\Delta_h$  es definida negativa por ser negativos sus autovalores, y en consecuencia, no singular.

Tras este breve estudio, podemos aproximar (2.2) por el siguiente problema discreto:

$$\left. \begin{aligned} &\text{Hallar } u_h \in \mathbb{R}^{N \times M} \text{ tal que} \\ &-\lambda L_h u_h + u_h = f_h \end{aligned} \right\} \quad (2.9)$$

donde  $(f_h)_k = f(x_i, y_j)$  con  $k = i + (1 - N)j$ ,  $i = 0, 1, \dots, N$ ,  $j = 0, 1, \dots, M$ , que tiene entonces solución única calculable mediante el método de Cholesky.

## Capítulo 3

# Resultados numéricos

### 3.1. Ejemplo de comprobación

Antes de utilizar la discretización obtenida para reducir el ruido en una imagen digital, vamos a comprobar con un ejemplo, que los cálculos son correctos.

Para ello, vamos a resolver el problema en ecuaciones en derivadas parciales (2.2) tomando como solución exacta definida en  $\Omega = [0, 1] \times [0, 1]$

$$u(x, y) = \sin^2(\pi x) \sin^2(\pi y),$$

que cumple la condición tipo Dirichlet  $u = 0$  sobre  $\partial\Omega$ .

Por lo tanto, tenemos que

$$\Delta u(x, y) = 2\pi^2 (\cos^2(\pi x) \sin^2(\pi y) + \sin^2(\pi x) \cos^2(\pi y) - 2 \sin^2(\pi x) \sin^2(\pi y)),$$

y, por tanto, para que se cumpla la ecuación, el segundo miembro debe ser

$$f(x, y) = -\lambda 2\pi^2 (\cos^2(\pi x) \sin^2(\pi y) + \sin^2(\pi x) \cos^2(\pi y) - 2 \sin^2(\pi x) \cos^2(\pi y)) + \sin^2(\pi x) \sin^2(\pi y).$$

Resolvamos ahora el sistema (2.9) con la discretización invariante por rotaciones, los resultados se muestran en la figura 3.1 a continuación.

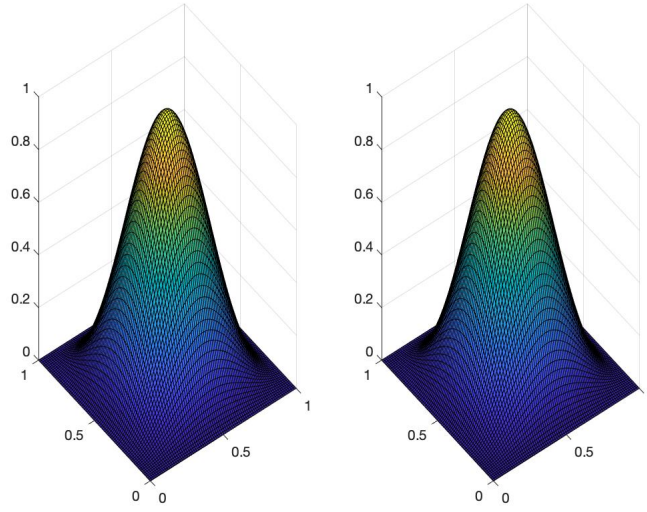


Figura 3.1: A la izquierda, la gráfica de  $u(x, y)$ , a la derecha, la solución aproximada.

Ahora, estudiemos el orden del error dado por  $e_h = u - u_h$ , y estudiémoslo tanto en la norma euclídea y la norma  $L^2$  discreta dada por

$$\|x\|_h = h\|x\|_2.$$

Asimismo, calcularemos también los errores relativos de la norma euclídea y la norma  $L^2$  discreta del error, los cuales vienen dados por

$$e_{r,2} = \frac{\|e_h\|_2}{\|u\|_2},$$

$$e_{r,h} = \frac{\|e_h\|_h}{\|u\|_h}.$$

En la tabla siguiente se muestran los valores de las normas euclídea y  $L^2$  discreta, y sus respectivos errores relativos.

$N$	$M$	$h$	$\ e_h\ _2$	$\ e_h\ _h$	$e_{r,2}$	$e_{r,h}$
10	10	$9,0909 \cdot 10^{-2}$	$1,6952 \cdot 10^{-1}$	$1,5411 \cdot 10^{-2}$	$3,5895 \cdot 10^{-3}$	$3,9485 \cdot 10^{-2}$
20	20	$4,7619 \cdot 10^{-2}$	$8,7364 \cdot 10^{-2}$	$4,1602 \cdot 10^{-3}$	$5,2252 \cdot 10^{-4}$	$1,0973 \cdot 10^{-2}$
40	40	$2,490 \cdot 10^{-2}$	$4,4544 \cdot 10^{-2}$	$1,0864 \cdot 10^{-3}$	$7,0460 \cdot 10^{-5}$	$2,8889 \cdot 10^{-3}$
80	80	$1,2346 \cdot 10^{-2}$	$2,2520 \cdot 10^{-2}$	$2,7802 \cdot 10^{-4}$	$9,1462 \cdot 10^{-6}$	$7,4084 \cdot 10^{-4}$

Tabla 3.1: Errores en norma euclídea y  $L^2$  discreta para distintos pasos  $h$ .

Comprobemos ahora, con ayuda de la tabla 3.1 y las gráficas en escala log-log de la figura 3.2 el orden numérico de los errores calculados.

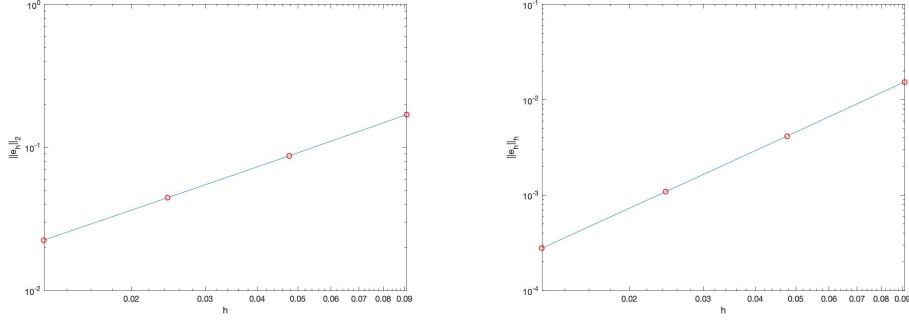


Figura 3.2: a la izquierda,  $\|e_h\|_2$  frente al paso, a la derecha  $\|e_h\|_h$  frente al paso.

Como podemos observar la figura 3.2, la pendiente de la norma euclídea es uno, y de la norma  $L^2$  discreta es dos, por lo tanto el orden del error de la norma euclídea del error es uno y de la norma  $L^2$  discreta es dos, como ya se puede intuir en la tabla 3.1.

## 3.2. Resultados con imágenes

Veamos ahora la resolución del problema discreto (2.9) donde  $f$  es la imagen original a la que le aplicaremos ruido de tipo gaussiano, sal y pimienta o uniforme. Denotaremos la imagen original por  $O$ , por  $f_h$  la imagen con ruido, y por  $u_h$  y  $v_h$  las imágenes tras aplicar la discretización del laplaciano invariante por rotaciones y el problema de reducción de ruido de cuarto orden abordado en [8], respectivamente.

El problema tratado en [8] consiste en resolver el siguiente problema en ecuaciones en derivadas parciales:

$$\left. \begin{aligned} &\text{Hallar } u : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R} \text{ tal que} \\ &\lambda \Delta^2 u + u = f \text{ en } \Omega \\ &u = 0 \text{ sobre } \partial\Omega \\ &\frac{\partial u}{\partial n} = 0 \text{ en } \partial\Omega \end{aligned} \right\}, \quad (3.1)$$

donde  $n = (n_1, n_2)$  es el vector normal orientado positivamente en  $\partial\Omega$ .

Tomemos primero la imagen de Lenna en escala de grises, y apliquemos ambos métodos.



Figura 3.3: Lenna en escala de grises.

Veamos los resultados obtenidos para los distintos tipos de ruido definidos en el primer capítulo. Empezamos con el ruido gaussiano.



Figura 3.4: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 5$ .



Figura 3.5: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 5$ .



Figura 3.6: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 1$ .



Figura 3.7: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 1$ .



Figura 3.8: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,5$ .



Figura 3.9: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,5$ .



Figura 3.10: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,25$ .



Figura 3.11: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,25$ .

Ahora, con ruido sal y pimienta.



Figura 3.12: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 5$ .



Figura 3.13: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 5$ .





Figura 3.14: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 1$ .



Figura 3.15: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 1$ .



Figura 3.16: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,5$ .



Figura 3.17: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,5$ .



Figura 3.18: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,25$ .

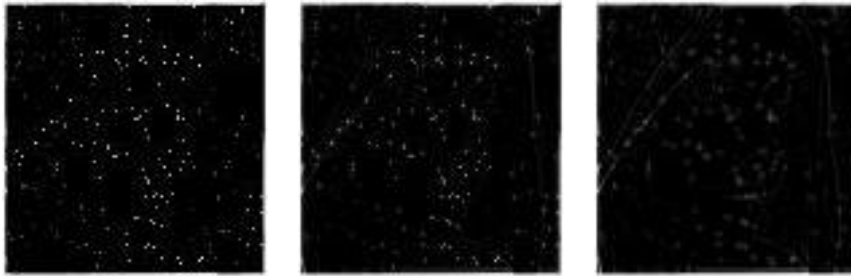


Figura 3.19: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,25$ .

Y por último, para ruido uniforme.



Figura 3.20: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 5$ .



Figura 3.21: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 5$ .



Figura 3.22: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 1$ .



Figura 3.23: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 1$ .



Figura 3.24: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,5$ .



Figura 3.25: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,5$ .



Figura 3.26: De izquierda a derecha,  $f_h$ ,  $u_h$  y  $v_h$  para  $\lambda = 0,25$ .



Figura 3.27: De izquierda a derecha,  $O - f_h$ ,  $O - u_h$  y  $O - v_h$  con  $\lambda = 0,25$ .

Como podemos comprobar después de estos primeros resultados, la discretización del laplaciano invariante por rotaciones, al eliminar el ruido, nos deja la imagen menos difuminada que en el otro caso. Esto se debe a que al resolver el problema en ecuaciones en derivadas parciales (3.1), la solución calculada es mucho más regular que en el problema (2.2), lo que no permite que haya saltos muy bruscos, y por eso la imagen resultante está más difuminada.

Vamos a ver ahora qué ocurre con la siguiente imagen de la figura 3.28.

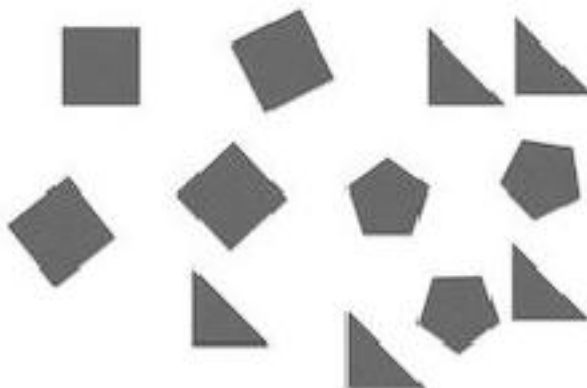


Figura 3.28: Imagen en escala de grises con distintas formas rotadas.

Como podemos ver en la imagen, hay varias formas rotadas, y con ello vamos a comprobar si es mejor la discretización del laplaciano invariante por rotaciones dada por (2.6), o si es mejor la discretización de trece puntos del bilaplaciano para el caso de imágenes con formas rotadas.

Empecemos viendo los resultados para ruido gaussiano.

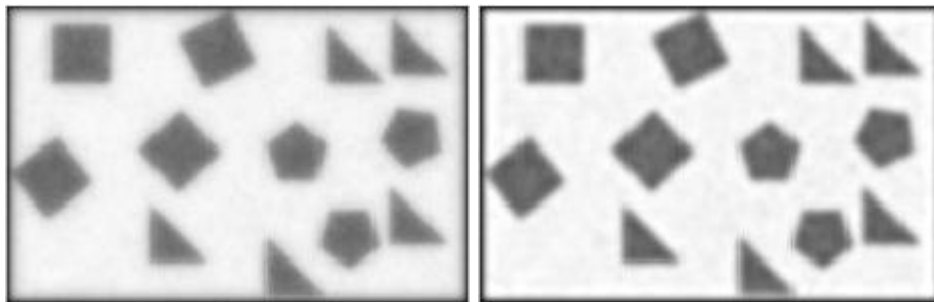
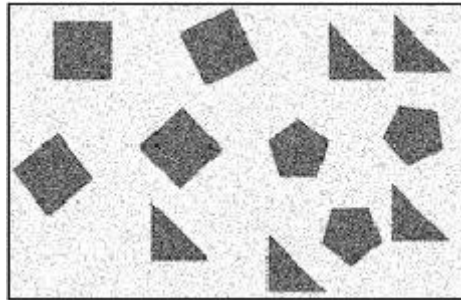


Figura 3.29: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .



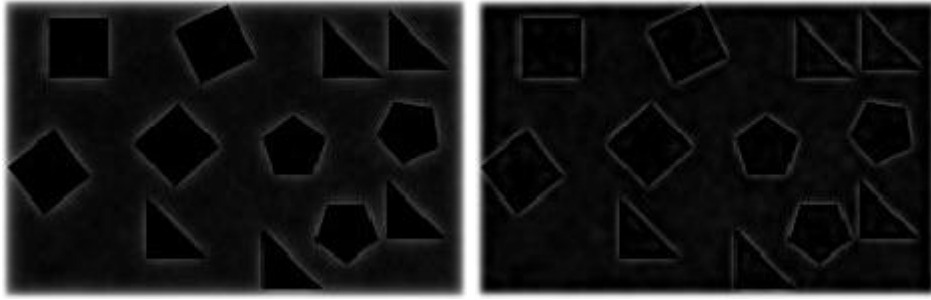


Figura 3.30: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .

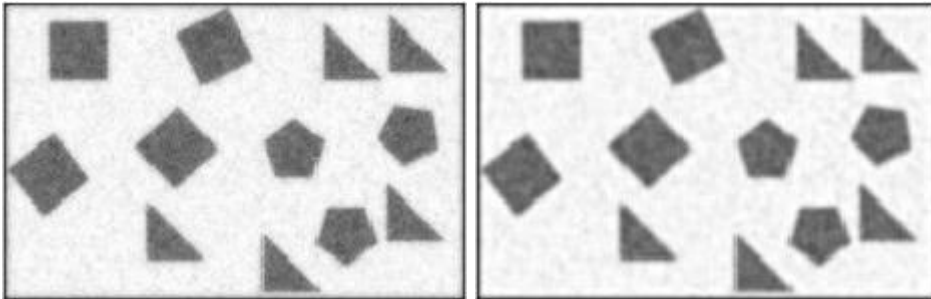
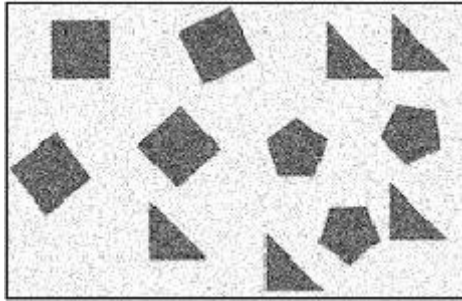


Figura 3.31: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .



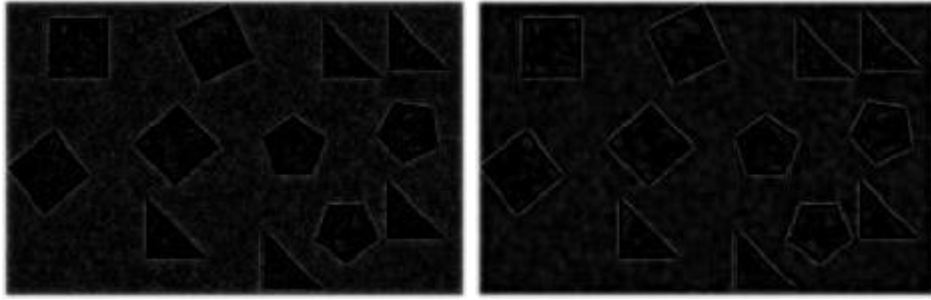


Figura 3.32: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .

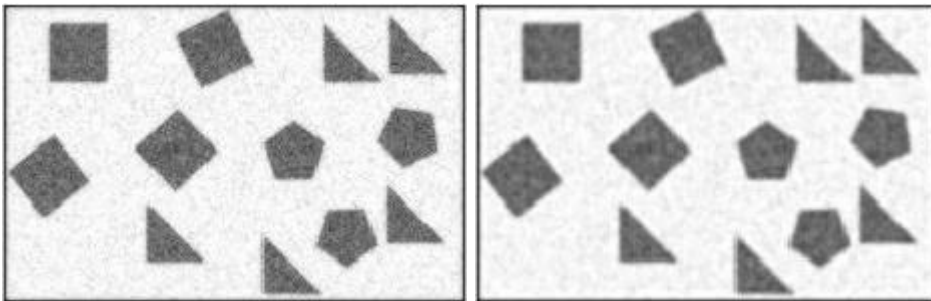
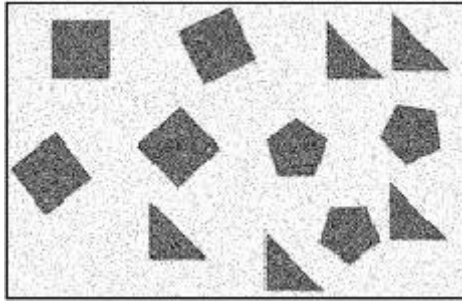


Figura 3.33: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .



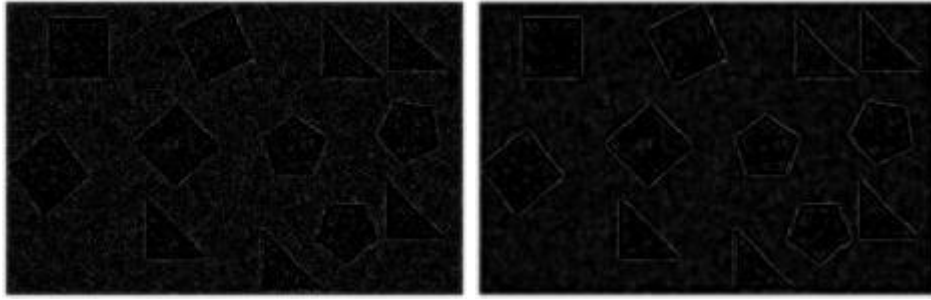


Figura 3.34: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .

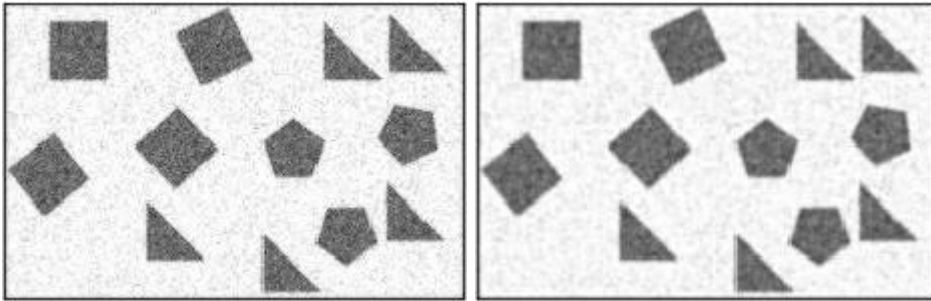
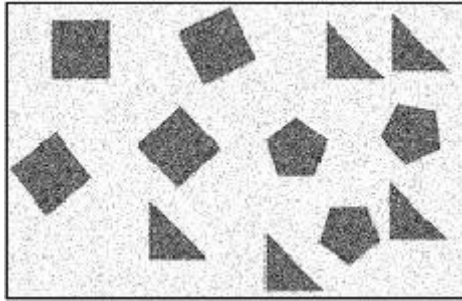


Figura 3.35: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .





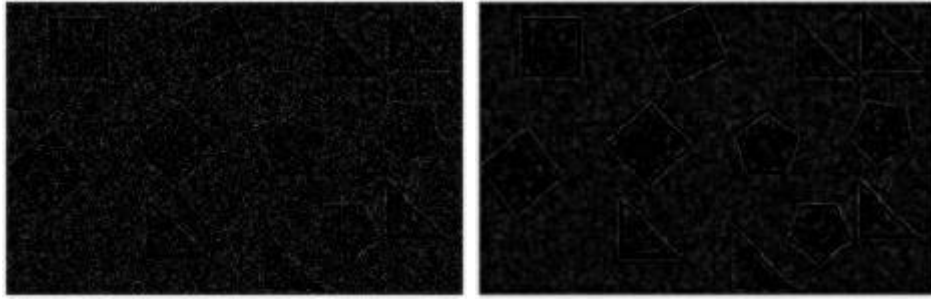


Figura 3.36: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .

Ahora, veremos qué ocurre con ruido sal y pimienta.

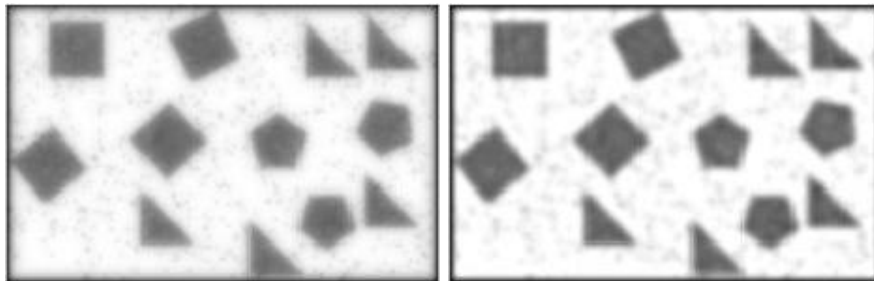
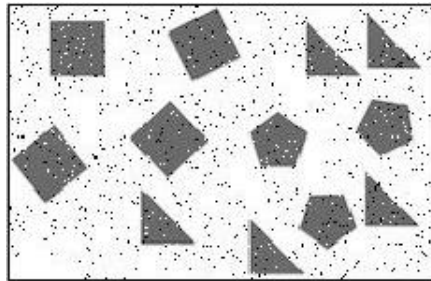


Figura 3.37: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .



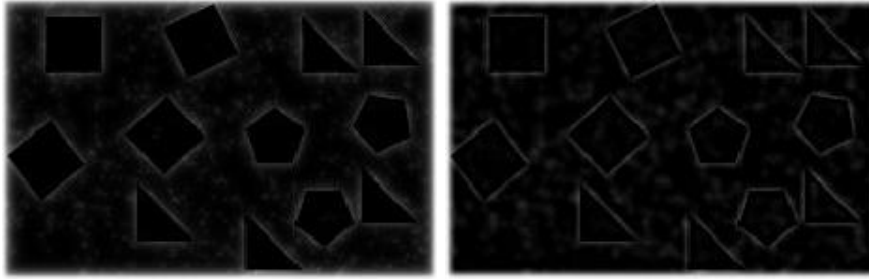


Figura 3.38: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .

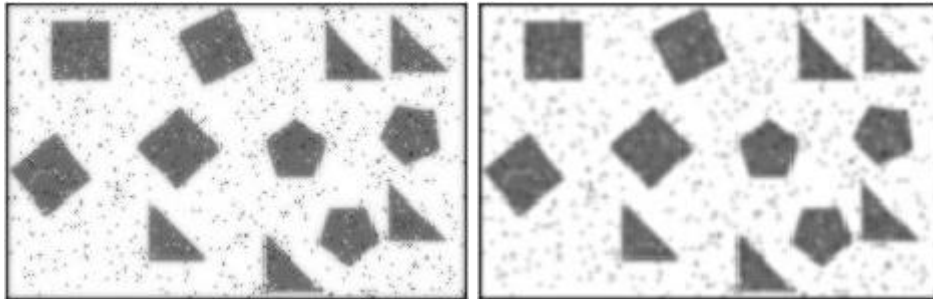
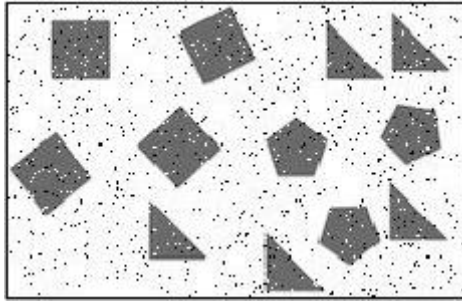


Figura 3.39: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .



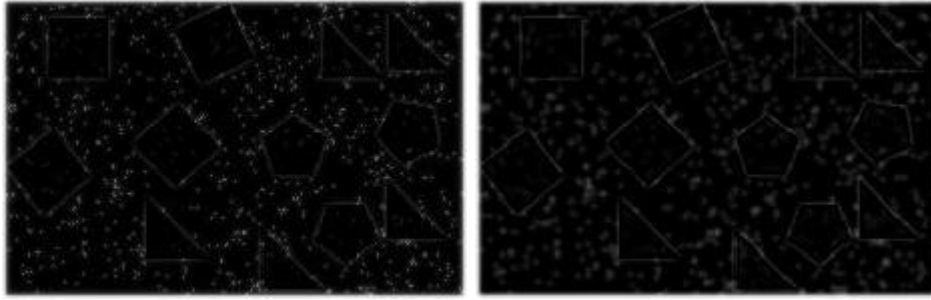


Figura 3.40: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .

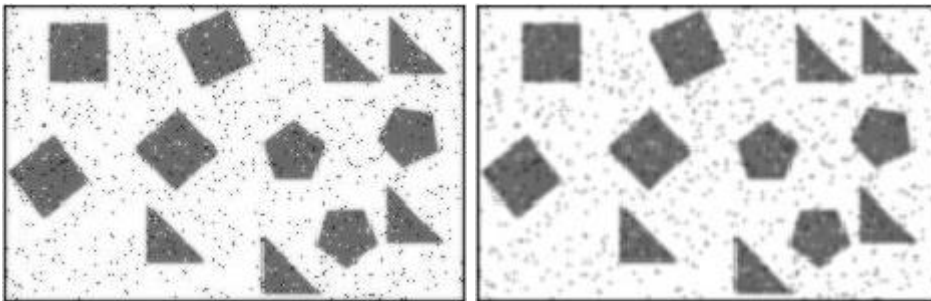
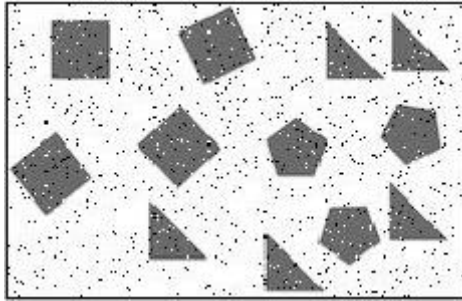


Figura 3.41: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .



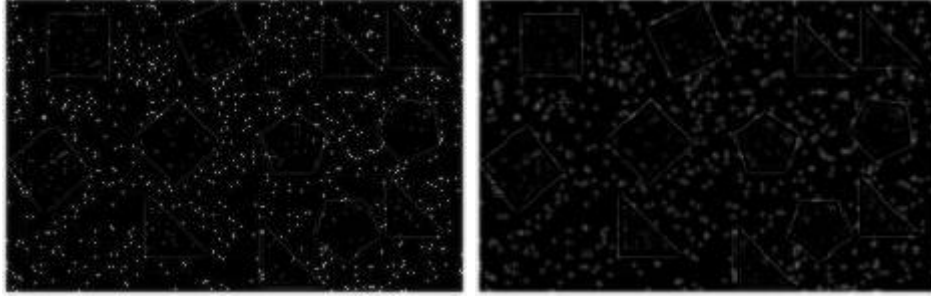


Figura 3.42: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .

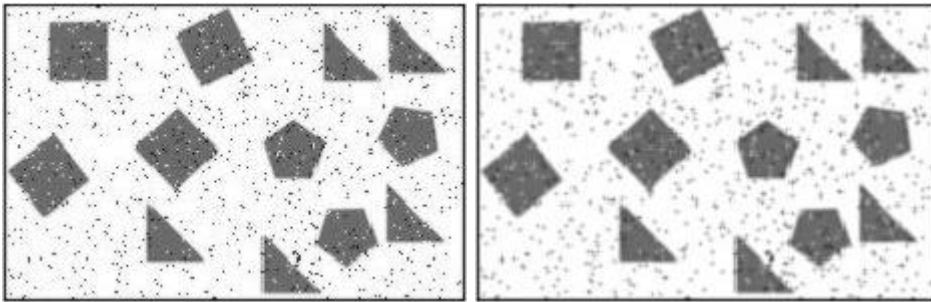
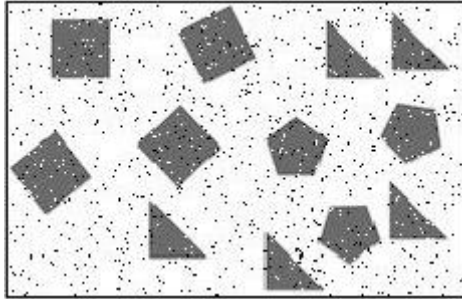


Figura 3.43: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .



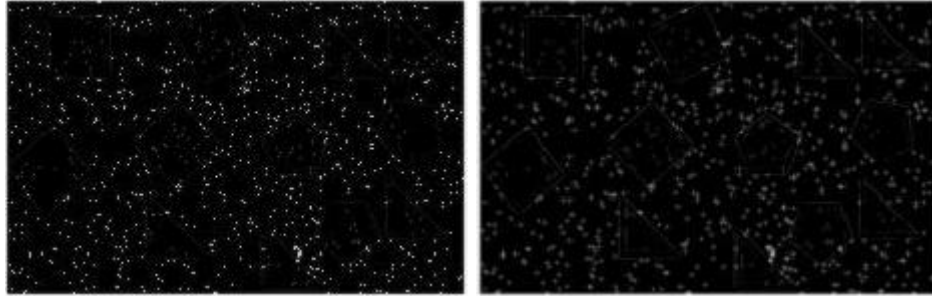


Figura 3.44: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .

Y por último, para el ruido uniforme, obtenemos lo siguiente.

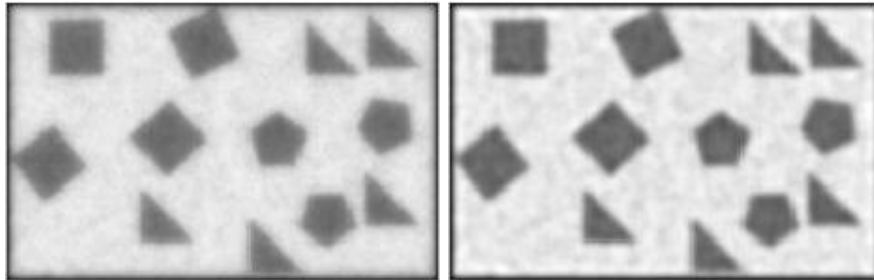
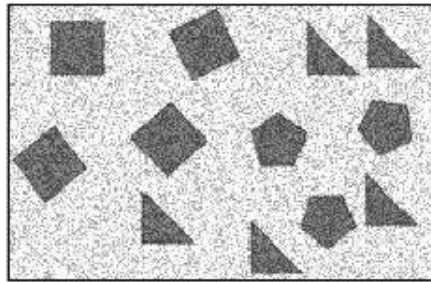


Figura 3.45: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .



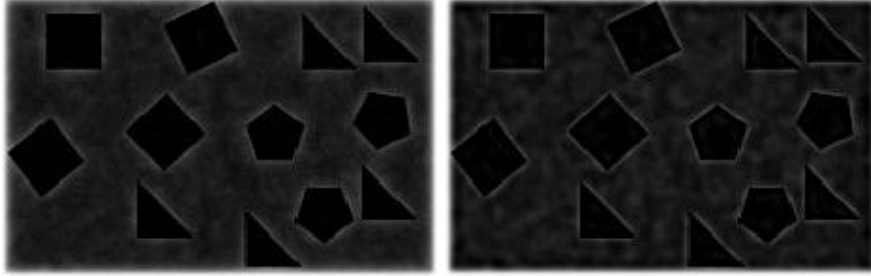


Figura 3.46: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 5$ .

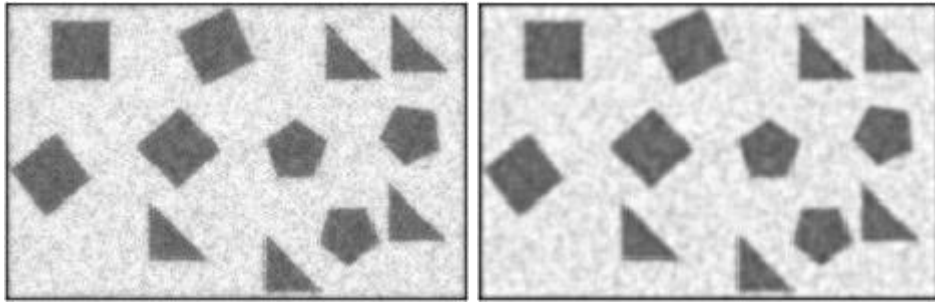
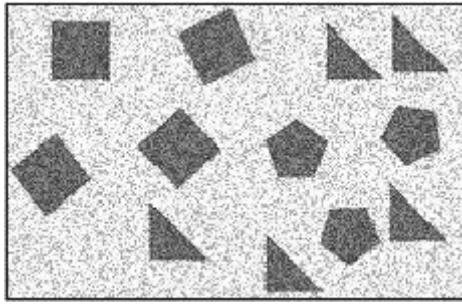


Figura 3.47: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .



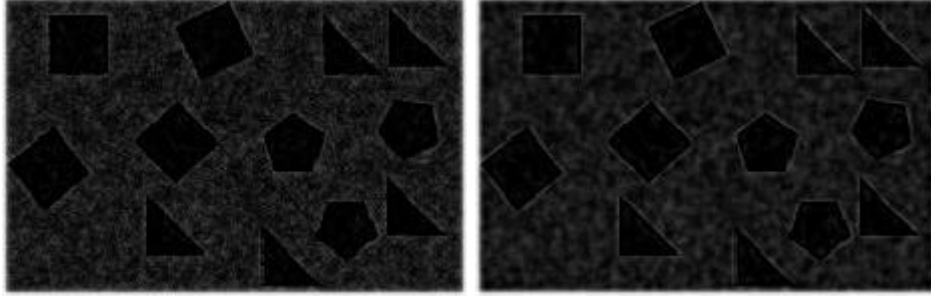


Figura 3.48: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 1$ .

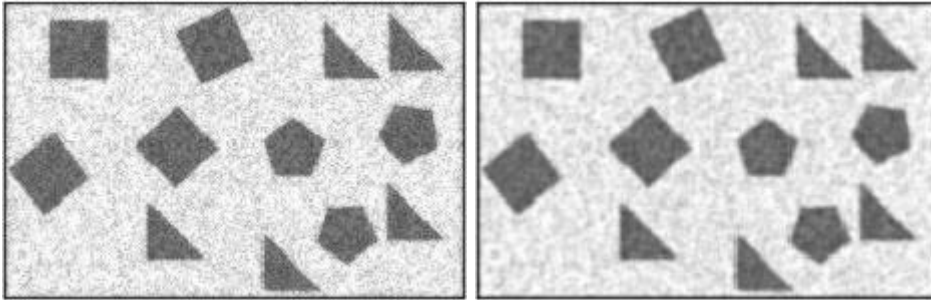
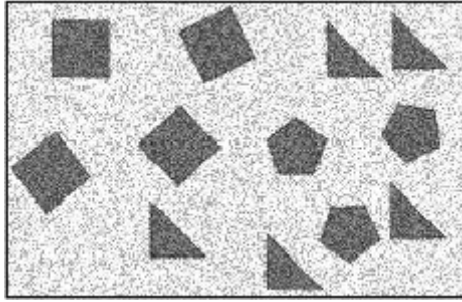
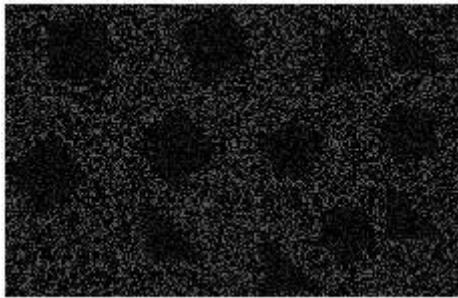


Figura 3.49: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .



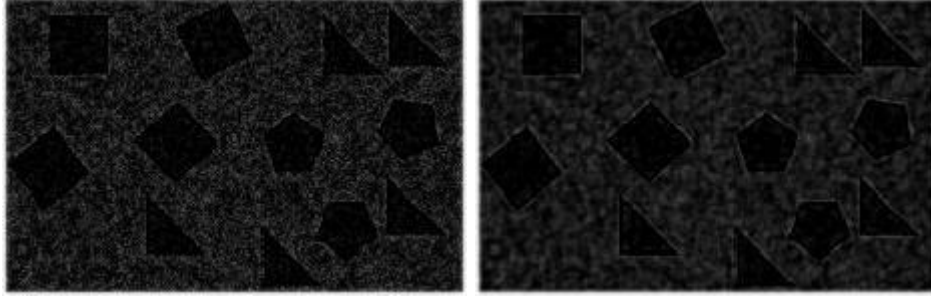


Figura 3.50: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,5$ .

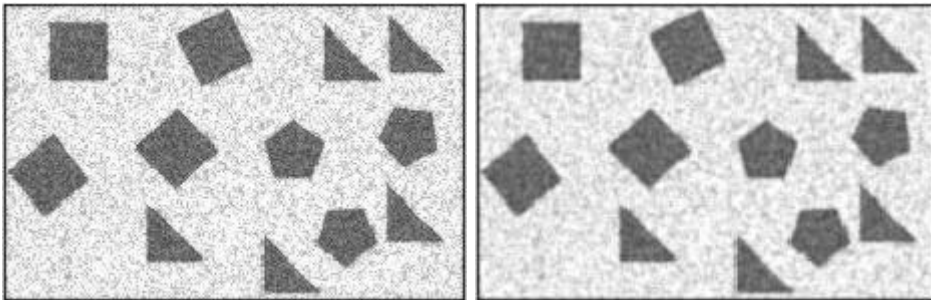
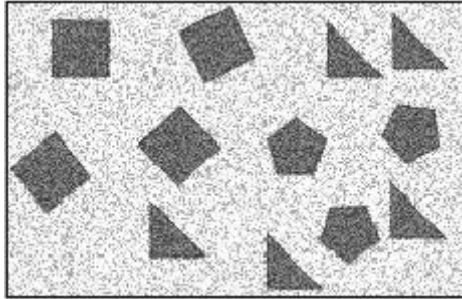
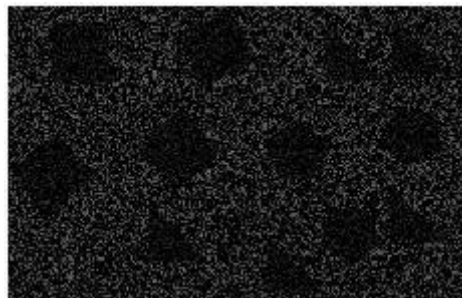


Figura 3.51: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .





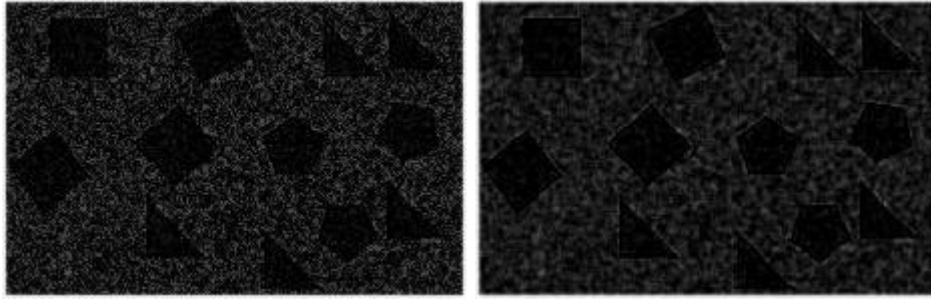


Figura 3.52: Arriba,  $f_h$ , abajo, de izquierda a derecha,  $u_h$  y  $v_h$ , con  $\lambda = 0,25$ .

Vamos a centrarnos ahora, particularmente, en los cuadrados y en los pentágonos, ya que estos están rotados.



Figura 3.53: Cuadrados de la figura 3.33,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.



Figura 3.54: Cuadrados de la figura 3.41,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.



Figura 3.55: Cuadrado de la figura 3.49,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.



Figura 3.56: Pentágonos de la figura 3.33,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.



Figura 3.57: Pentágonos de la figura 3.41,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.



Figura 3.58: Pentágonos de la figura 3.49,  $f_h$ ,  $u_h$  y  $v_h$  respectivamente.

Como podemos observar, los resultados obtenidos tras resolver el problema (2.2) no solo nos proporciona una imagen menos difuminada, si no que también conserva mejor los bordes.

# Bibliografía

- [1] Evans, *Partial Differential Equations*, 2nd ed., American Mathematical Soc., 2010.
- [2] Lui, S. H., *Numerical Analysis of Partial Differential Equations* 1st ed., Wiley, 2011.
- [3] Gordon Dennis Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 3rd ed., Oxford Applied Mathematics and Computing Science Series, 1978.
- [4] Brezis, H., *Análisis funcional*, 1st edition, Alianza editorial, Madrid, 1984.
- [5] Solomon, C., Breckon, T., *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, 1st ed, Wiley-Blackwell, 2011.
- [6] Zhu, W., Chan, T., *Image Denoising Using Mean Curvature of Image Surface*, Society for Industrial and Applied Mathematics, 2012.
- [7] <https://en.wikipedia.org/wiki/Lenna>.
- [8] Ferreira Doval, Nuno. *Estudio e implementación de un algoritmo para reducir el ruido en imágenes bidimensionales*. Trabajo de Fin de Grado del Grado de Matemáticas. Universidad de Santiago de Compostela, Santiago de Compostela, 2018.
- [9] González, R. and Woods, R., *Digital Image Processing*, 4th ed., Pearson education, 2017.



# Apéndices



# Programa ejemplo de comprobación

```
clear all
close all
%%%%%%%%

% Declaracion de variables
global lambda
% Introduccion de los datos
lambda=1;
% Numero de puntos de la discretizacion
nx=20;
ny=nx;
% Paso
h=1/(nx+1);
% Mallado
x=0:h:1;
y=0:h:1;
% Inicializacion
U=zeros(nx+2,ny+2);
% Calculo de la solucion aproximada
U(2:nx+1,2:ny+1)=lapinv(x(2:nx+1),y(2:ny+1));
% Calculo de la solucion exacta
U_ex=uex(x(:),y);
% Representacion grafica de las soluciones exacta y aproximada
figure
subplot(1,2,1)
surf(x,y,U), title('Solucion aproximada')
axis([0 1 0 1 0 1])
```

```
subplot(1,2,2)
surf(x,y,U_ex), title('Solucion exacta')
axis([0 1 0 1 0 1])
% Calculo de los errores
e1=norm(U,2); % Norma euclidea de la solucion aproximada
e2=norm(U-U_ex,2); % Norma euclidea del error
eh=h*e2; % Norma L2 discreta
erh=e2/e1; % Error relativo en norma euclidea
er2=h*er; % Error relativo en norma discreta
fprintf('paso %e \n',h)
fprintf('norma euclidea del error %e \n',e2)
fprintf('norma euclidea del error relativo %e \n',er2)
fprintf('norma L2 discreta del error %e \n',eh)
fprintf('norma L2 discreta del error relativo %e \n', erh)
```



# Programa para la eliminación de ruido

```
clear all
close all
%%%%%%%%

% Declaracion de variables
global lambda
lambda=0.25;
% Lectura de la imagen
% I=imread('formas.jpg'); % Imagen con las formas rotadas
I=imread('Lena.png'); % Imagen de Lenna
O=rgb2gray(I); % Conversion a escala de grises
O=im2double(G);
% Guardamos la imagen en otra matriz para mantener la imagen con original
R=O;
% Adicion del ruido
R=imnoise(R,'gaussian'); % Ruido gaussiano
%R=imnoise(R,'salt & pepper'); % Ruido sal y pimienta
%R=imnoise(R,'speckle'); % Ruido uniforme
% Datos
[nx,ny]=size(R);
h=1;
% Adicion de un marco negro a la imagen
R(1,:)=zeros(1,ny);
R(:,1)=zeros(nx,1);
R(nx,:)=zeros(1,ny);
```

```

R(:,ny)=zeros(nx,1);
% Construimos las matrices de las discretizaciones
Lh=matinv(nx,ny,h); % Matriz asociada al laplaciano discreto
H=mattre(nx,ny,h); % Matriz asociada al bilaplaciano discreto
% Escribimos la imagen con ruido f_h como una columna
b=zeros(nx*ny,1);
for l=1:nx
    for m=1:ny
        b((m-1)*nx+1)=R(l,m);
    end
end
% Obtenemos la imagen con ruido reducido al resolver el sistema
fh=Lh\b;
ffh=H\b;
% Imagen con ruido reducido mediante el laplaciano discreto
A=zeros(nx,ny);
for l=1:nx
    for m=1:ny
        A(l,m)=fh((m-1)*nx+1);
    end
end
% Imagen con ruido reducido mediante el bilaplaciano discreto
AA=zeros(nx,ny);
for l=1:nx
    for m=1:ny
        AA(l,m)=ffh(l+(m-1)*nx);
    end
end
% Obtencion de las imagenes del error
N1=O-R;
N2=O-A;
N3=O-AA;
% Representacion de las imagenes
figure
subplot(2,3,1)
imshow(R), title('f_h')

```

```
subplot(2,3,2)
imshow(A), title('u_h')
subplot(2,3,3)
imshow(AA), title('v_h')
subplot(2,3,4)
imshow(N1), title('O-f_h')
subplot(2,3,5)
imshow(N2), title('O-u_h')
subplot(2,3,6)
imshow(N3), title('O-v_h')
```



# Función para calcular la solución del ejemplo de comprobación

```
function UU=lapinv(xx,yy)
% Funcion que calcula la solucion numerica del problema
%% Variables de entrada
% xx, yy: vector del mallado del espacio
%% Variable de salida
% UU: matriz con la solucion aproximada

% Declaracion de variables
global lambda
% Datos
h=diff(xx(1:2)); % Paso
LLx=length(xx); % Numero de puntos en el mallado
LLy=length(yy); % Numero de puuntos en el mallado
L=LLx*LLy; % Orden de la matriz
% Lado derecho del sistema
b=zeros(1,L);
for i=1:LLx
    for j=1:LLy
        b((j-1)*LLx+i)=f(xx(i),yy(j));
    end
end
% Calculo de la matriz del sistema
% Matriz T
dT=sparse(1:LLx,1:LLx,-8,LLx,LLx); % Diagonal
uT=sparse(1:LLx-1,2:LLx,1,LLx,LLx); % Super diagonal
```

```

lT=sparse(2:LLx,1:LLx-1,1,LLx,LLx); % Subdiagonal
T=dT+uT+lT;
% Matriz C
dC=sparse(1:LLx,1:LLx,1,LLx,LLx); % Diagonal
uC=sparse(1:LLx-1,2:LLx,1,LLx,LLx); % Superdiagonal
lC=sparse(2:LLx,1:LLx-1,1,LLx,LLx); % Subdiagonal
C=dC+uC+lC;
% Inicializacion
A=sparse(L);
% Construcción de la primera fila de la matriz del sistema
A(1:LLx,1:LLx)=T;
A(1:LLx,(LLx+1):2*LLx)=C;
% Construcción de la fila i-esima
for i=2:LLy-1
    A((LLx*(i-1)+1):i*LLx,(LLx*(i-2)+1):(i-1)*LLx)=C;
    A((LLx*(i-1)+1):i*LLx,(LLx*(i-1)+1):i*LLx)=T;
    A((LLx*(i-1)+1):i*LLx,(LLx*i+1):(i+1)*LLx)=C;
end
% Construcción de la última fila
A((LLy-1)*LLx+1:LLx*LLy,((LLy-2)*LLx+1):(LLy-1)*LLx)=C;
A((LLy-1)*LLx+1:LLx*LLy,((LLy-1)*LLx+1):LLy*LLx)=T;
% Matriz final
A=eye(L)-lambda*A/(3*h^2);
% Solución del sistema
sol=A\b';
% Escribimos la solución del sistema en forma de matriz
UU=zeros(LLx,LLy);
for i=1:LLx
    for j=1:LLy
        UU(i,j)=sol(i+(j-1)*LLx);
    end
end
end
end

```

# Función de construcción de la matriz del laplaciano discreto

```
function Lh=matinv(nx,ny,h)
% Funcion que construye la matriz asociada al laplaciano discreto
%% Variables de entrada
% nx: Numero de filas de la imagen
% ny: Numero de columnas de la imagen
% h: Paso de la discretizacion
%% Variables de salida
% Lh: matriz asociada al laplaciano discreto

% Declaracion de variables
global lambda
% Datos
h2=h^2;
n=nx*ny;
% Construcción de la matriz T
dT=sparse(1:nx,1:nx,-8/h2,nx,nx); % Diagonal
uT=sparse(1:nx-1,2:nx,1/h2,nx,nx); % Superdiagonal
lT=sparse(2:nx,1:nx-1,1/h2,nx,nx); % Subdiagonal
T=dT+uT+lT;
% Contrucción de la matriz C
dC=sparse(1:nx,1:nx,1/h2,nx,nx); % Diagonal
uC=sparse(1:nx-1,2:nx,1/h2,nx,nx); % Superdiagonal
lC=sparse(2:nx,1:nx-1,1/h2,nx,nx); % Subdiagonal
C=dC+uC+lC;
% Inicializamos la matriz
```

```

Lh=sparse(n);
% Construcción de la primera fila
Lh(1:nx,1:nx)=T;
Lh(1:nx,(nx+1):2*nx)=C;
% Construcción de la fila i-esima
for i=2:ny-1
    Lh((nx*(i-1)+1):i*nx,(nx*(i-2)+1):(i-1)*nx)=C;
    Lh((nx*(i-1)+1):i*nx,(nx*(i-1)+1):i*nx)=T;
    Lh((nx*(i-1)+1):i*nx,(nx*i+1):(i+1)*nx)=C;
end
% Contrucción de la ultima fila
Lh((ny-1)*nx+1:nx*ny,((ny-2)*nx+1):(ny-1)*nx)=C;
Lh((ny-1)*nx+1:nx*ny,((ny-1)*nx+1):ny*nx)=T;
% Matriz final
Lh=eye(n)-lambda*Lh/(3*h2);
end

```



# Función de construcción de la matriz del bilaplaciano discreto

```
function H=mattre(nx,ny,h)
% Funcion que contruye la matriz asociada al bilaplaciano discreto
%% Variables de entrada
% nx: Numero de columnas de la imagen
% ny: Numero de filas de la imagen
% h: Paso de la discretizacion
%% Variables de salida
% H: matriz asociada al bilaplaciano discreto de 13 puntos

% Declaracion de variables
global lambda
% Datos
n=nx*ny;
h4=h^4;

% Construccion de la matriz B1
B1d=sparse(1:nx,1:nx,21,nx,nx);
B1ud=sparse(1:nx-1,2:nx,-8,nx,nx);
B1ld=sparse(2:nx,1:nx-1,-8,nx,nx);
B1uud=sparse(3:nx,1:nx-2,1,nx,nx);
B1lld=sparse(1:nx-2,3:nx,1,nx,nx);
B1=(B1d+B1ud+B1ld+B1uud+B1lld)/h4;
B1(1,1)=22/h4;
B1(nx,nx)=22/h4;
% Construccion de la matriz B
```

```

Bd=sparse (1:nx,1:nx,20,nx,nx);
Bud=sparse (1:nx-1,2:nx,-8,nx,nx);
Bld=sparse (2:nx,1:nx-1,-8,nx,nx);
Buud=sparse (3:nx,1:nx-2,1,nx,nx);
Blld=sparse (1:nx-2,3:nx,1,nx,nx);
B=(Bd+Bud+Bld+Buud+Blld)/h4;
B(1,1)=21/h4;
B(nx,nx)=21/h4;
% Construcccion de la matriz C
Cd=sparse (1:nx,1:nx,-8,nx,nx);
Cud=sparse (1:nx-1,2:nx,2,nx,nx);
Cld=sparse (2:nx,1:nx-1,2,nx,nx);
C=(Cd+Cud+Cld)/h4;
% Construcccion de la matriz I
I=sparse (1:nx,1:nx,1/h4,nx,nx);

% Construcccion de la matriz
% Inicializamos la matriz
H=sparse(n);
% Construcccion de la primera fila
H(1:nx,1:nx)=B1;
H(1:nx,(nx+1):2*nx)=C;
H(1:nx,(2*nx+1):3*nx)=I;
% Construcccion de la segunda fila
H((nx+1):2*nx,1:nx)=C;
H((nx+1):2*nx,(nx+1):2*nx)=B;
H((nx+1):2*nx,(2*nx+1):3*nx)=C;
H((nx+1):2*nx,(3*nx+1):4*nx)=I;
% Construcccion de la fila k-esima
for k=3:ny-2
    H(((k-1)*nx+1):k*nx,((k-3)*nx+1):((k-2)*nx))=I;
    H(((k-1)*nx+1):k*nx,((k-2)*nx+1):((k-1)*nx))=C;
    H(((k-1)*nx+1):k*nx,((k-1)*nx+1):(k*nx))=B;
    H(((k-1)*nx+1):k*nx,(k*nx+1):((k+1)*nx))=C;
    H(((k-1)*nx+1):k*nx,((k+1)*nx+1):((k+2)*nx))=I;
end

```

```

% Construcción de la fila (L-1)-ésima
H(((ny-2)*nx+1):(ny-1)*nx, ((ny-4)*nx+1):(ny-3)*nx)=I;
H(((ny-2)*nx+1):(ny-1)*nx, ((ny-3)*nx+1):(ny-2)*nx)=C;
H(((ny-2)*nx+1):(ny-1)*nx, ((ny-2)*nx+1):(ny-1)*nx)=B;
H(((ny-2)*nx+1):(ny-1)*nx, ((ny-1)*nx+1):ny*nx)=C;
% Construcción de la fila L-ésima
H(((ny-1)*nx+1):ny*nx, ((ny-3)*nx+1):(ny-2)*nx)=I;
H(((ny-1)*nx+1):ny*nx, ((ny-2)*nx+1):(ny-1)*nx)=C;
H(((ny-1)*nx+1):ny*nx, ((ny-1)*nx+1):ny*nx)=B1;
% Matriz final
H=H+eye(n);
end

```



# Funciones ejemplo

## Función solución

```
function z=f(x,y)
% Funcion que calcula el valor de la funcion f en el punto (x,y)

global lambda
z= -lambda*2*pi^2*(cos(pi*x).^2.*sin(pi.*y).^2-...
    2*sin(pi.*x).^2.*sin(pi.*y).^2+sin(pi*x).^2.*cos(pi.*y).^2)+...
    sin(pi*x).^2.*sin(pi.*y).^2;
end
```

## Solución exacta

```
function z=uex(x,y)
% Funcion que calcula la solucion exacta del problema en el punto (x,y)

z=sin(pi*x).^2.*sin(pi*y).^2;
end
```